Dynamic management policies embedded digital control systems

JAVIER VÁSQUEZ
University of Costa Rica, Computer Science Deparment
San José, Costa Rica
jvasquez@ecci.ucr.ac.cr
JOSE L. VÁSQUEZ
University of Costa Rica, Sede del Atlántico
Cartago, Costa Rica
jose.vasquez@ucr.ac.cr
CARLOS M. TRAVIESO
University of Las Palmas de Gran Canaria, Signals and Communications Department.
Las Palmas de Gran Canaria, España.
ctravieso@dsc.ulpgc.es, http://www.gpds.ulpgc.es

Abstract : In this paper we present a model that enables us to separate the control rules from the embedded systems. This allows for better adaptation to changing user needs, and better use of data generated in the control devices, since it facilitates interoperability with other enterprise systems. By separating the control of the devices, the proposed model provides a broad longevity to the components of digital systems that implement it, provides greater flexibility to adapt the response of the devices with changing administrative interests, and promises a decrease in costs maintenance, since it reduces the need to modify the software embedded in the control device. To illustrate the use of this model, we implemented a monitoring application using a software architecture of 3 layers.

Key-words: Dynamics rules, distributed control, embedded systems, 3-Their Architecture, NIST-DCM

1 Introduction

Advances in technology, low cost of microcontrollers and devices, such as sensors and actuators, and the proliferation of Internet connections had led to the emergence of digital system controller (DSC) that are designed to monitoring and controlling equipment distributed by remote human intervention. These DSC consists of a series of sensors controlled commonly locally by a microcontroller. Even though there may be work distributed and it is usual that the controller intelligence is permanently associated with each of the interconnected devices.

For purposes of this article we mean by *control* the attempt to impose predictability unsafe entities that react to events and we understand *control policy* the predefined set of events and must be activated when so determined by the control devices.

To carry out the distribution of control is proposed to model distributed control (MCD) that maintains separate political control devices, hence any change in such policies do not necessarily alter the hardware or software used in such DSC. This method allows for incremental construction of control devices become more sophisticated and that the useful life of these increases, because of the possibility of reuse in the context of policies more in line with future needs of customers.

In the MCD to maintain separate policies of mechanisms implements the concept of control rule which is complemented by the use of a platform for distributed applications based on client server architecture and protocol stack TCP / IP.

2 Background

In the late twentieth and early twenty-first has been a great interest to the research on sensor networks, perhaps due to the confluence of three factors:

- Decreasing cost of technology needed to produce digital systems.
- Gradual increase in miniaturization and capabilities of such systems (which most often are wireless)
- Widespread use of embedded systems [2]

Some domains and potential uses are: everyday applications (heating, security, resource allocation, location of cars and people), industrial manufacturing, transport applications (aviation, automobiles, power distribution, prevention and solution of traffic congestion), banking applications (debit and credit cards), health applications (medical monitors, medical diagnostics), monitoring applications (species, tools, environmental conditions, events), environmental applications (diseases of crops, natural resources) and other [1, 2, 3, 4, 8, 11, 12].

Advances in technology, making it likely that the very short term, current limitations that have “motes”[1] are irrelevant, and that we can avoid worrying too much by the interconnection of sensor networks reaching. In this regard, in [1] presents a new “mote” design by Intel Company, called Stargate with tens of megabytes RAM, gigabytes of persistent memory and ability to interact with Internet. In [6] considering the use of chips of small size and large capacity for handling video and audio as well as its transmission in compact format, e.g., TC35273XB by Toshiba. Also define a processing architecture of three levels, processors near where video cameras will be equipped with databases, filtering techniques, compression and pattern recognition. Devices that use these chips will approach and turning capabilities of each camera and shall be fitted with solar cells to become self sufficient. Similarly, in [11] is advocated because it gives local video processing in base stations to reduce the amount of data sent over the channel.

Regarding the management of control rules, in [3] proposes the use of responsible control laws to ensure respect the preferences of users of sensor networks. The use of virtual machines to control management has been proposed in [9], those seeking to reduce the work of code conversion and thus the power consumption of devices. This virtual machine has 3 levels of abstraction *Handlers,* responding to system events; *Operations*, that are functional execution units and finally the *Code Capsules*, which are units of spreading code. The intercommunication between sensors has also arrived at structuring and delegating to agents, in [14] proposes to handle sensor networks that hypothetically could be composed of

[1] Devices that integrate sensors with computation and comunication.

thousands of sensors arranged hierarchically, in equipment that could be handled by simple agents.

In [12] proposes an architecture similar to that described in this article, e.g. Internet use, use of "base stations", which for us would be the device, also say they have used Postgres to storage data. Similarly, in [5] describes a security application for a building, using a platform consisting of multiple motion sensors, sending messages through GSM and reception in a PDA, using the UDP protocol for sending data and the TCP protocol to send administrative information.

In order to adapt and update applications dynamically, in [10] proposes the use of an intermediate layer (*middleware*) which frees the hardware from user needs. It is in this layer where they intend to be given the automatic adjustment of parameters that simplify distributed processing of data from mobile wireless sensor. We see that the *middleware* has been given a very poor use, and that is only used essentially for parameterization. In [7], also proposes the use of an intermediate layer, to address each sensor, based on the priority of it at the time of the event.

Like previous efforts, NIST-DCM model [13] proposed in 1999 by the U.S. government to control distributed administration, also makes a clear difference between what the system must perform and how to do so. This leads to rigid patterns of control in the area of monitoring, which once defined the policies, they are assembled and distributed to devices, so any changes to these policies, involves modifying the software resident on the devices, hardware, or both, reducing their useful lives.

3 Distributed Control Model (DCM)

The DSC market are themselves imbedded in control, thereby inhibiting use one different from that for which they were conceived. In this paper we identify the benefits to be gained by changing this habit. We propose the use of a three-tier architecture that allows abstract control policies and that the DSC can dynamically modify the composition and behavior. It omits any consideration about the type of sensors used and the problems associated with the possible topologies of sensor network, focusing only on the application layer to propose a specific solution to the problem of handling dynamic control.

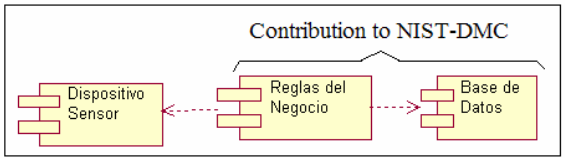

Fig. 1 Arquitecture for the control scheme with distributed intelligence.

Figure 1 shows the 3-tier architecture used in the proposed model for distributed dynamic control (referred to as DCM). In the *customer layer* are placed monitoring devices (Device Sensor), implemented through logic circuits. In the *business layer* are Web services that contain business rules. There is a web server, servlet, jsp and java beans, here are dynamic control rules, detected events are processed and shoot the message to their local or remote actuators provided in the control rule; in the *server layer* is the database that can store data on players, devices, policies and events. It is also reflected in this figure that the NIST-DMC-model lacking the support for control policies can benefit from these two layers.

The following sections describe the main components of the DCM: control devices, dynamic management of control rules and distributed management of devices.

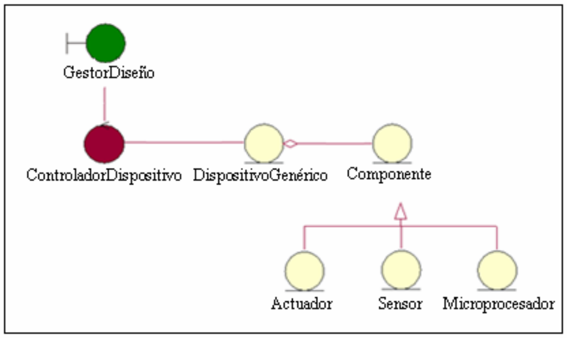

Fig. 2 Devices design with reusable components.

3.1 Control Device Characterization

Figure 2 shows that control devices are designed using the MVC pattern, allowing abstract interface for managing devices, the distributed controller and the generic device model created by adding persistent compounds, grouped in 3 categories: the *sensors* that come to shape the set of artifacts that are permanently recorded the occurrence of events, *actuators* that come to be the set of artifacts that react before certain policies, instructed to do so by a driver, and finally are *microprocessors*, microcontrollers that can be or even personal computers.

From the abstract point of view, any device in the MCD is modeled as an object that has a number of attributes and that exhibit a behavior. In this context, the components of a device is seen as the attributes of the same. Additionally, the DCM behavior of the object is determined by its operations, and in the case of a device, the most common operations are related to the query and management of sensors and actuators, providing a service and setting its mode of operation.

In designing a device as an object, encapsulation property allows construction details are hidden from one device, for use requiring only that the device has a set of operations or commands to interact with its. With encapsulation the system gains flexibility and allows a better way to adapt to the changes experienced by the devices, as well as changes in user requirements. To integrate a device into the system, besides its activation requires that this be recorded in the database along with control policies in which it will intervene. Once operational the device can be accessed and managed remotely.

3.2 Facility Management Control Rules

A control rule consists of one or more events and a series of actions that give them an answer. Any event generated by the sensors is recorded in a log, is evaluated against the control rules and if necessary conditions are fulfilled, the respective shares are issued they are, so to speak, the methods associated with a device capable of interpreting and carry them out. These actions may in turn be events of different control rules. It is specialist work of the configuration of control devices and the definition of basic control rules. These can be edited by the administrator of DSC, which allows distinguishing two instances to manage them:

1. As part of the assembly of components, is defined and stored in persistent memory in the form of policies, a default behavior. Once installed, the DSC, the administrator can define a set of control rules, which modify the behavior. Rules should be evaluated and approved by the security administrator.

2. Whenever the DSC comes into operation as a first step checks for updates to the set of rules, In this case, they replace those predefined in the persistent memory and will be in effect until you import a new set of them. The predefined rules are necessary to ensure a minimum performance in case you can not communicate with the controller policy.

3.3 Distributed Architecture for Dynamic Policy Management

The management control rules principally consist of two phases, one of which allows dynamic modification of those rules of control and the other controls the activation of the same.

Modification of control rules: the DCM allows remote users with security privileges to do so, to edit a set of policies on devices. Being residents control policies in a table in the database, allowing an adjustment in line with the interests and capabilities of the administrator, making these devices have a longest life and are reusable.

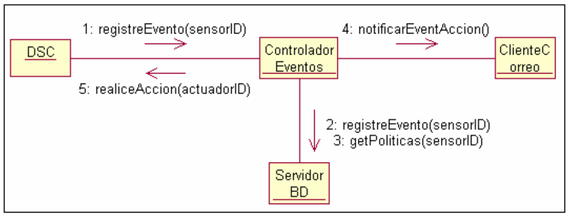
1: registreEvento(sensorID)
4: notificarEventAccion()
DSC
Controlador Eventos
ClienteC orreo
5: realiceAccion(actuadorID)
2: registreEvento(sensorID)
3: getPoliticas(sensorID)
Servidor BD

Fig. 3 Mechanism to implement the control rules.

Activation of the control rules: when a device driver communicates to the occurrence of any event, it must be registered in the database. Also, check if you have associated a control rule, if so the controller extracts the orders database, and instructs the corresponding devices to perform the actions indicated.

Figure 3 describes the protocol of activation of rules within DCM:

1. The DSC communicates to Controller Events the fulfilling of any condition.
2. The Controller Events recorded in the database the received message.
3. The Controller Events consults into database to determine if there is associated a control rule, if so, obtains from database actions to be performed.
4. The Controller Events could notify the user via the mail client has been established
5. The Controller Events orders the different actors perform the corresponding actions that can be conceptualized as events in chain.

A device only performs the actions associated with an event when these are indicated by Controller Events. However, sometimes the device can perform an operation on its own initiative, a case is when the device loses communication with the controller.

3.4 Example: Designing a Security Manager

The intrusion warning systems have wide distribution. These systems consist of multiple sensors, such as opening doors or windows, motion detectors, heat, temperature, smoke and glass breakage. The actuators are usually used in DSC are these speakers, lights and video cameras. We will configure the DSC with opening sensors, and temperature, and as actuators only lights, a webcam and a phone connection. As a starting point, define a consistent policy on conventional horn activate when a sensor is activated. It could also be other conventional policies such as:
Policy 1: If the sensor becomes active opening, then the alarm sounds.
Policy 2: If the sensor temperature exceeds 50 degrees, then the alarm sounds

Policy 3: If the motion detector is triggered, then send a message to specific phone number

This rigid set of policies will make the system more vulnerable burglar to one who knows them beforehand. With DCM can be altered at will the system behavior, at any given time redefining the set of policies. For example could be added to the system behavior against intruders these rules:
Policy 1: If the motion detector is activated at the instant of time t, turn on lights and activate the webcam.
Policy 2: If the sensor temperature exceeds 40 degrees in time t + α, send message to specific phone number

For this system could be implemented was necessary to provide the DSC with remote interfaces to authenticate users, modify the behavior of the devices in a dynamic and coordinate with the mail server associated.

4 Implementation

We developed a prototype security system that includes all the elements described in the proposed model. In this implementation is free software tools used. One of the prototype monitoring devices comprises a general purpose microcontroller that has an ethernet interface built into the chip, facilitating their connection to the network. The other components of monitoring device is a temperature sensor, a sensor of locks and two electrical switches for lights and air conditioning. The controller is a Java application, which is primarily responsible for verifying and implementing the current control at the moment. The database, implemented with MySQL, contains the entities and relationships needed to show the performance characteristics of the model. In this case include users, devices, log of events and control rules. Finally, we used Tomcat as web application container, consisting of XHTML pages, EJB, JSP, and Servlets. This application provides the user a platform to query and control rules specify the database, and communicate directly with the monitoring device.

Once developed and implemented the DCM, there have been three important results:
1. Ability to alter runtime behavior of a DSC
2. Integration of open source technologies in the design and implementation of embedded systems
3. DCM_NIST model was increased to enable distribution of dynamic control.

5 Conclusions

- The modeling of a monitoring device through the concept of object increases its level of abstraction, which facilitates the maintenance of their hardware and software.. Also, the definition of an interface that hides the implementation details contributes to the interoperability and scalability of the system.
- The physical separation between policy and control devices enables the reuse of devices and policies.
- The incorporation of the Controller Events in the DCM as a component of the mechanism of control rules relieves the monitoring devices work event processing and communication with the database server.
- The DCM gives system administrators a tool for remote management of their policies and devices, making it ideal for remote monitoring or under very critical conditions due to cost or very few time for this.
- The DCM enhances collaboration between microcontrollers, because through control policies and the IP

address associated with each device simplifies the delegation of tasks regardless of whether it is a homogeneous or mixed architecture.

- By defining the control rules based on events and actions, there is a better use of data generated by events, because the data can be leveraged both within the DSC and other DSC and applications connected via the network.

REFERENCES

[1] Conner, W. S.; Heidemann, J.; Krishnamurthy, L.; Wang , X. & Yarvis, M. **"Workplace Applications of Sensor Networks"**, Intel Research and Development University of Southern California, Information Sciences Institute. USC/ISI Technical Report ISI-TR-2004-591 http://www.isi.edu/div7/publication_files/Conner04a.pdf

[2] Culler, D.; Berkeley, D. & Srivastava, E. M. **"Overview of Sensor Networks"**, IEEE Computer , vol:37 , no:8 , pp:41-49. Agosto 2004

[3] Deshpande, A.; Guestrin, C. & Madden, S. R. **"Resource - AwareWireless Sensor - Actuator Networks"**, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. 2005

[4] Dunkels, A.; Alonso, J.; Voigt, T.; Ritter, H. & Schiller, J. **"Connecting Wireless Sensornets with TCP/IP Networks"**, Proceedings of the Second International Conference on Wired/Wireless Internet Communications (WWIC2004), Frankfurt (Oder), Germany, February 2004

[5] Dunkels, A.; Feeney, L.M.; Grönvall, B. & Voigt, T. **"An integrated approach to developing sensor network solutions"**, Proceedings of the Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA, antes IEEE SNPA) Boston, Massachusetts, USA, August 2004 http://www.sics.se/~adam/sanpa2004.pdf

[6] Feng, W.; Walpole, J.; Feng, W. & Pu, C.. **"Moving Towards Massively Scalable Video-Based Sensor Networks"**, Proceedings of the 9th annual international conference on Mobile computing and networking. ACM Press New York, NY, USA. 2003

[7] Heinzelman, B.; Murphy, L.; Carvalho, H. & Perillo, M. **"Middleware to Support Sensor Network Applications"**, IEEE Network Magazine Special Issue. Jan. 2004.

[8] Kumar, S.; Shepherd, D. & Zhao, F. Guest Editors. **"Collaborative Signal and Information Processing in Micro-Sensor Networks"**, IEEE Signal Processing Magazine, March 2002

[9] Levis, P.; Gay, D. & Culler, D. **"Active Sensor Networks"**, Proceedings of the 2nd USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI), May 2005

[10] Liu, T. & Martonosi, M. **"Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems"**, ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPoPP), June 2003

[11] Obraczka, K.; Manduchi, R. & Garcia–Luna, J.J. **“Managing the Information Flow in Visual Sensor Networks”,** 5th International Symposium on Wireless Personal Multimedia Communications, Honolulu, HI, October 2002. http://www.soe.ucsc.edu/~manduchi/Papers/178-obraczka.pdf

[12] Szewczyk, R.; Polastre, J.; Mainwaring, A. & Culler, D. **“Lessons From A Sensor Network Expedition”**, European Workshop on Wireless Sensor Networks 2004 http://www.eecs.harvard.edu/~mdw/course/cs263/papers/gdi-ewsn04.pdf

[13] Schneeman, Richard D. **“Implementing a Standards-based Distributed Measurement and Control Application on the Internet”**, U.S. Department of Commerce, 1999.

[14] Yadgar, O.; Kraus, S. & Ortiz, C.L. **“Scaling-up distributed sensor networks: cooperative large-scale mobile-agent organizations”**, http://www.umiacs.umd.edu/~sarit/Articles/bar-ilan-chapter.pdf